individual points of contact made by the object (step **1310**). It computes a centroid, or other average point, of the determined points of contact (step **1320**). Touch program **520** then computes a standard deviation of the centroid as well as the variance (step **1330**), and determines the pointer size based on the centroid and the standard deviation (step **1340**). These computations are preferably performed on a real-time basis to provide immediate system response to the touch input. In order to achieve optimum results and accuracy, analog touch controller **460** preferably generates 150 points per second or more. Touch program **520** may also use the amount of pressure imposed on touch screen panel **474** as a function of time to determine the size of object. As shown in FIG. **14**A, for example, if the amount of pressure increases or decreases sharply at a particular instant in time, touch point program **520** may determine that the touch corresponds to a pen. A finger touch, on the other hand, results in a gradual increase and decrease in pressure as illustrated by a smoother curve in FIG. **14**B

FIG. **13**A is a flowchart illustrating the process of determining the size of the object making contact with the viewing area, and implementing selected functions based on the determined size, in accordance with the embodiment described with reference to FIGS. **14**A and **14**B. Touch point program **520** first detects and determines the individual points of contact made by the object (step **1310**A). As described above, touch controller **460** generates at least **150** points per second in order to more accurately detect contact points on the viewable area. Touch program **520** then determines the amount of pressure imposed on the viewable area by the object (step **1320**A), and the amount of time the object makes contact with the viewable area (step **1330**A). The pressure and time data is utilized by program **520** to compute the pointer size of the object, similar to the exemplary process described in FIGS. **14**A and **14**B, above (step **1340**A).

Program **510** can also be programmed to correlate certain pointer size to certain objects and invoke corresponding functions or tools (step **1350**). Such GUI provides a richer, yet simplified interaction between the user and mobile telephone **310**. If program **510** determines that the pointer size of the object corresponds to the size of a finger (step **1350**A), program **510** may initiate a navigation tool (Step **1360**A). If the pointer size corresponds to the size of several fingers, program **510** may invoke a drag function of the navigation tool. On the other hand, if program **510** determines that the pointer size of the object corresponds to size of a sharp point or pen (Step **1350**A), program **510** may initiate a drawing tool supported by drawing program **540** (step **1370**A). Similarly, if program **510** determines that the pointer size of the object corresponds to size of a pencil eraser (step **1350**A), program **510** may initiate an erase function of the drawing tool (step **1380**A). One skilled in the art may easily vary the functions or tools initiated by program **510**. Additionally, the functions or tools may be commercial software packages, predetermined functions, or user-defined macros.

In addition to using the pointer size to determine the desired GUI, program **510** can also incorporate other characteristics of the user touch, e.g., gestures or movements, to simplify GUI and maximize screen real estate. A gesture recognizing interface extends the ability of the present invention to distinguish between different sized pointers to track gestures and movement of user input based on vector direction and magnitude, all in the context of active user application. This type of contextual gesture interface can infer by context, the implement, and the gesture chosen by the user what functions the user wishes to invoke. Accordingly, all these functions are available without menus or scroll bars and do not require additional screen areas to display the functions.

Program **510** recognizes other characteristics of the touch input including the context of the input, namely the task or sub-task applications running when the GUI is invoked. If a user is in a document navigation application, for example, program **510** interprets a quick drag to the right as a next page function. If the underlying task is an editing application, program **510** may interpret the same gesture as a highlight function and highlight a portion of the document touched by the user. Similarly, in graphics application, a quick drag to the right may invoke a drawing tool to draw from the starting point to the ending point of the touch points. In a document viewing application, the same touch may invoke a navigation tool to move the view of the document in the direction of the finger drag.

All of the above functions and features described above focuses on providing intuitive GUIs and minimize the need for users to memorize complicated, hierarchical menus or procedures. Additionally, the present invention maximize available screen real estate while providing a wide array of GUI and tools.

It will be apparent to those skilled in the art that various modifications and variations can be made in the system of the present invention and in construction of this system without departing from the scope or spirit of the invention. Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. The specification and examples should be considered as exemplary only, with the true scope and spirit of the invention indicated by the following claims.

What is claimed is:

1. A method of providing a touch-responsive user interface comprising the steps of:

detecting an object making contact with a physical viewing area;

determining a pointer size of the object, based on a rate of change in the amount of pressure imposed by the object on the physical viewing area; and

activating a function corresponding to the pointer size,

wherein the determining step includes the substep of: <determining the pointer size corresponding to a finger; and

wherein the activating step includes the substep of: activating a navigation tool in response to the determined pointer size.

2. The method of claim **1**, wherein the determining step further includes the substep of:

determining the pointer size corresponding to a pen, and

wherein the activating step further includes the substep of: activating a draw function in response to the determined pointer size corresponding to a pen.

3. The method of claim **1**, wherein the determining step further includes the substep of:

determining the pointer size corresponding to several fingers, and

wherein the activating step further includes the substep of: activating a drag function in response to the determined pointer size corresponding to several fingers.

4. The method of claim **1**, wherein the rate of change in the amount of pressure imposed by the object on the physical viewing area indicates that the amount of pressure changed gradually over a predetermined range of time.